

Maceda, Rafael  
2016-01337  
CoE 197S

## Feature Activity 01

### A. Pulling an Image

1. Let's see what images we have currently on our machine, by running `'docker images'`:
2. On a fresh Docker installation, we should have no images. Let's pull one from Dockerhub.

```
C:\Users\Rafael>docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
C:\Users\Rafael>docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu
61e03ba1d414: Pull complete
4afb39f216bd: Pull complete
e489abdc9f90: Pull complete
999fff7bcc24: Pull complete
Digest: sha256:6aab78d1825b4c15c159fecc62b8eef4fd0c693a15aace3a605ad44e5e2df0c
Status: Downloaded newer image for ubuntu:16.04
docker.io/library/ubuntu:16.04
```

3. We can also pull different versions on the same image.

```
C:\Users\Rafael>docker pull ubuntu:16.10
16.10: Pulling from library/ubuntu
dca7be20e546: Pull complete
40bca54f5968: Pull complete
61464f23390e: Pull complete
d99f0bcd5dc8: Pull complete
120db6f90955: Pull complete
Digest: sha256:8dc9652808dc091400d7d5983949043a9f9c7132b15c14814275d25f94bca18a
Status: Downloaded newer image for ubuntu:16.10
docker.io/library/ubuntu:16.10
```

Then when we run `'docker images'` again, we should get:

```
C:\Users\Rafael>docker images
REPOSITORY      TAG              IMAGE ID         CREATED          SIZE
ubuntu          16.04           065cf14a189c    19 hours ago    135MB
ubuntu          16.10           7d3f705d307c    3 years ago     107MB
```

4. Over time, your machine can collect a lot of images, so it's nice to remove unwanted images.

Run `'docker rmi <IMAGE ID>'` to remove the Ubuntu 16.10 image we won't be using.

```
C:\Users\Rafael>docker rmi 7d3f705d307c
Untagged: ubuntu:16.10
Untagged: ubuntu@sha256:8dc9652808dc091400d7d5983949043a9f9c7132b15c14814275d25f94bca18a
Deleted: sha256:7d3f705d307c7c225398e04d4c4f8512f64eb8a65959a1fb4514dfde18a047e7
Deleted: sha256:d9db289b9342d9617596cd6ee3bba988629e24d9afa5db4e4b0e4e491c65007d
Deleted: sha256:a87725e8597b97f2399bc3aa50b0e2eec903b8ce19055668d3befb012918205c
Deleted: sha256:38cf10a2801529348366953e9b933d3524360dedc91d3e4d5d7f941da0c973c9
Deleted: sha256:61172966249d43026dbd017eec3a9575e37bddf8a269a9f09ecb559d7bfe7fef
Deleted: sha256:57145c01eb80040fdd0a24cde20af4788605b49593188d4f7efab099af89a08e
```

Alternatively, you can delete images by tag or by a partial image ID. In the previous example, the following would have been equivalent:

```
- `docker rmi 31`
- `docker rmi ubuntu:16.10`
```

Running `docker images` should reflect the deleted image.

```
C:\Users\Rafael>docker rmi 7d3f705d307c
Untagged: ubuntu:16.10
Untagged: ubuntu@sha256:8dc9652808dc091400d7d5983949043a9f9c7132b15c14814275d25f94bca18a
Deleted: sha256:7d3f705d307c7c225398e04d4c4f8512f64eb8a65959a1fb4514dfde18a047e7
Deleted: sha256:d9db289b9342d9617596cd6ee3bba988629e24d9afa5db4e4b0e4e491c65007d
Deleted: sha256:a87725e8597b97f2399bc3aa50b0e2eec903b8ce19055668d3befb012918205c
Deleted: sha256:38cf10a2801529348366953e9b933d3524360dedc91d3e4d5d7f941da0c973c9
Deleted: sha256:61172966249d43026dbd017eec3a9575e37bddf8a269a9f09ecb559d7bfe7fef
Deleted: sha256:57145c01eb80040fdd0a24cde20af4788605b49593188d4f7efab099af89a08e

C:\Users\Rafael>docker images
REPOSITORY    TAG        IMAGE ID      CREATED       SIZE
ubuntu        16.04      065cf14a189c  19 hours ago  135MB

C:\Users\Rafael>
```

## B. Running a Container

1. Using the Ubuntu 16.04 image we downloaded, we can run a our first container. Unlike a traditional virtualization framework like VirtualBox or VMWare, we can't just start a virtual machine running this image without anything else: we have to give it a command to run.

The command can be anything you want, as long as it exists on the image. In the case of the Ubuntu image, it's a Linux kernel with many of the typical applications you'd find in a basic Linux environment.

1. Let's do a very simple example. Run `docker run ubuntu:16.04 /bin/echo 'Hello world! '`

```
C:\Users\Rafael>docker run ubuntu:16.04 /bin/echo 'Hello world!'
'Hello world!'
```

2. Let's check what containers we have after running this. Run ``docker ps``

```
C:\Users\Rafael>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
```

That's strange: no containers right? The ``ps`` command doesn't show stopped containers by default, add the ``-a`` flag.

```
C:\Users\Rafael>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
b587dc5becea   ubuntu:16.04  "/bin/echo 'Hello wo..."  3 minutes ago   Exited (0) 3 minutes ago          epic_haibt
```

3. Let's do something a bit more interactive. Run ``docker run ubuntu:16.04 /bin/bash``

```
C:\Users\Rafael>docker run ubuntu:16.04 /bin/bash

C:\Users\Rafael>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS          NAMES
84b0ad83f4f7   ubuntu:16.04  "/bin/bash"             13 seconds ago   Exited (0) 12 seconds ago          focused_ma
tsumoto
b587dc5becea   ubuntu:16.04  "/bin/echo 'Hello wo..."  5 minutes ago   Exited (0) 5 minutes ago          epic_haibt
C:\Users\Rafael>
```

The container exited instantly. Why? We were running the ``/bin/bash`` command, which is an interactive program. However, the ``docker run`` command doesn't run interactively by default, therefore the ``/bin/bash`` command exited, and the container stopped.

Instead, let's add the ``-it`` flags, which tells Docker to run the command interactively with your terminal.

```
C:\Users\Rafael>docker run -it ubuntu:16.04 /bin/bash
root@75623437d080:/#
```

You can now use this like a normal Linux shell. Try ``pwd`` and ``ls`` to look at the file system.

You can type ``exit`` to end the BASH session, terminating the command and stopping the container.

```
C:\Users\Rafael>docker run -it ubuntu:16.04 /bin/bash
root@75623437d080:/# pwd
/
root@75623437d080:/# ls
bin boot dev etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var
root@75623437d080:/# exit
exit
C:\Users\Rafael>
```

4. By default your terminal remains attached to the container when you run ``docker run``. What if you don't want to remain attached?

By adding the ``-d`` flag, we can run in detached mode, meaning the container will continue to run as long as the command is, but it won't print the output.

Let's run ``/bin/sleep 3600``, which will run the container idly for 1 hour:

```
C:\Users\Rafael>docker run -d ubuntu:16.04 /bin/sleep 3600
2b1d457b52eb7ba8fb973d802e4b12ce18fd6256e318d2a7bc999cf653c75ffb

C:\Users\Rafael>docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
2b1d457b52eb	ubuntu:16.04	"/bin/sleep 3600"	10 seconds ago	Up 9 seconds		priceless_galois

5. Now that the container is running in the background, what if we want to reattach to it?

Conceivably, if this were something like a web server or other process where we might like to inspect logs while it runs, it'd be useful to run something on the container without interrupting the current process.

To this end, there is another command, called ``docker exec``. ``docker exec`` runs a command within a container that is already running. It works exactly like ``docker run``, except instead of taking an image ID, it takes a container ID.

This makes the ``docker exec`` command useful for tailing logs, or "SSHing" into an active container.

Let's do that now, running the following, passing the first few characters of the container ID:

```
C:\Users\Rafael>docker exec -it 2b1 /bin/bash
root@2b1d457b52eb:/# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.0   4372   648 ?        Ss   19:00    0:00 /bin/sleep 3600
root         7  0.0  0.0  18232  3228 pts/0    Ss   19:04    0:00 /bin/bash
root        17  0.0  0.0  34420  2796 pts/0    R+   19:04    0:00 ps aux
root@2b1d457b52eb:/# exit
exit

C:\Users\Rafael>docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS   NAMES
2b1d457b52eb   ubuntu:16.04 "/bin/sleep 3600"        4 minutes ago   Up 4 minutes   priceless_galois
```

6. Instead of waiting 1 hour for this command to stop (and the container exit), what if we'd like to stop the Docker container now?

To that end, we have the `docker stop` and the `docker kill` commands. The prior is a graceful stop, whereas the latter is a forceful one.

Let's use `docker stop`, passing it the first few characters of the container name we want to stop.

```
C:\Users\Rafael>docker stop 2b1
2b1

C:\Users\Rafael>docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS   NAMES
2b1d457b52eb   ubuntu:16.04 "/bin/sleep 3600"        6 minutes ago   Exited (137) 24 seconds ago   priceless_galois
75623437d080   ubuntu:16.04 "/bin/bash"             10 minutes ago   Exited (0) 9 minutes ago      student_lehmann
84b0ad83f4f7   ubuntu:16.04 "/bin/bash"             12 minutes ago   Exited (0) 12 minutes ago      focused_matsumoto
b587dc5becea   ubuntu:16.04 "/bin/echo 'Hello wo..." 18 minutes ago   Exited (0) 18 minutes ago      epic_bt

C:\Users\Rafael>
```

## C. Removing Containers

From our previous example, we can see with `docker ps -a` that we have a container hanging around.

To remove this we can use the `docker rm` command which removes stopped containers.

```
C:\Users\Rafael>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
75623437d080   ubuntu:16.04   "/bin/bash"             15 minutes ago Exited (0) 14 minutes ago          stupefied_
lehmann
84b0ad83f4f7   ubuntu:16.04   "/bin/bash"             17 minutes ago Exited (0) 17 minutes ago          focused_ma
tsumoto
b587dc5becea   ubuntu:16.04   "/bin/echo 'Hello wo..." 22 minutes ago Exited (0) 22 minutes ago          epic_haibt

C:\Users\Rafael>docker rm 7562
7562

C:\Users\Rafael>docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS          PORTS          NAMES
84b0ad83f4f7   ubuntu:16.04   "/bin/bash"             17 minutes ago Exited (0) 17 minutes ago          focused_ma
tsumoto
b587dc5becea   ubuntu:16.04   "/bin/echo 'Hello wo..." 23 minutes ago Exited (0) 23 minutes ago          epic_haibt

C:\Users\Rafael>
```